Project Best Practice

This document outlines the best practices for creating projects in CGI Studio.

Platform selection and considerations

The first topic of selecting a platform is to decide if 3D is necessary or not. All platforms support 2D but not all platforms support 3D. The decision for 3D has already an impact on the design possibilities and resources which are needed. Is a 3D designer needed or just 2D designer? For 3D design Autodesk 3ds Max, Autodesk Maya or Blender are usually used. For 2D design Adobe Photoshop is the first choice, but some customers use different 2D design tools (Adobe InDesign, Figma, plain images, ...).
Depending on your decision, experts/designers for 2D and 3D are needed.
The second topic of selecting a platform is to decide about the necessary display resolution and number of supported displays. Is more than one display required or not? Not all platforms support more than 1 display. Regarding resolutions it must be clarified if resolution is supported by the platform. Higher resolutions could exceed the limit of some platforms. Some platforms have multi-layer support (layer blending is possible) and some platforms support just one layer. In best case the design is made in a way to benefit from the multi-layer support of a platform.

How to design an HMI

The design of an HMI should be done in a way to best fit to the platform specific abilities.

- Multi-Layer support
- Available blend modes. Not every platform supports all blend modes.
- Available BitmapProfiles (RGBA8888, RGB555, RLE, RLA, ...)

This document cannot cover all details about specific design topics, but there are some general rules.

- Keep images as small as possible
- Group functionalities/objects in design tools (avoid flat structures of design objects)
- Use meaningful names for objects and groups (avoid Layer1, Layer2, Layer3, ...)
- Avoid special effects which cannot be achieved by CGI Studio (maybe bake effects in images)
- Minimize the number of fonts used. For some use cases pre-rendered numbers/letters are a better approach.

Please also refer to the following detailed documentation.

- In this documentation

    - Quick Start Guide
    - Content Design Best Practices
    - Smart Importer Best Practices
    - Getting Started on Android

- *\cgi_studio_docs\Documentation*
    - CGI-Studio_2DEffects_SWDS.pdf
    - CGI-Studio_Layouter_SWDS.pdf

- *\cgi_studio_docs*
    - NXP_RT117X_Hardware_Setup_and_Usage_Guide_for_CGI-Studio.pdf
    - TelechipsTC8030_Dolphins_Plus_Hardware_Setup_Guide.pdf

In an early stage of the project, it must be decided how many languages need to be supported. Specifically, if you want to support RTL (Right To Left) languages. This must be considered in the design, and it must be specified how content is changing in case of RTL languages. Introducing RTL languages at later stages of the project can be considered as a new project in an existing project.

How to set up a team

The team should contain people of the following groups, beside management functions.

Software developer

C++ developers which understand the CGI framework and how to implement new behaviors and adapt the application code. It is also beneficial if SW developers understand CMake and Python. CMake is used to configure and extend the build system. Python is used to implement helper functionality for different purposes.

Technical artist

Technical artists can compose the graphical scenery in CGI SceneComposer. They import the graphical content provided by the designers and set up the scenery. They add behaviors and Controls to create the necessary functionality. By using the available set of controls and behaviors they can create new controls and BBBs (Behavior Building Blocks).

Designer

Designers should work closely together with technical artists to understand what's needed to implement some use cases and adapt the design accordingly.
A design fitting the needs of platform and CGI Studio environment can help to save memory and decrease GPU and CPU load.
Best team would be a team where the members of the group have knowledge in more than one area as this guarantees a good understanding of the other areas.
e.g. technical artists are also capable of doing debugging and C++ developers can also work in

Scenecomposer.

Translator

It is only necessary if multi language support is needed.
Usually, the activity of translating is outsourced to specialized external companies, but some customers do translations also in-house. CGI Studio has the possibility to export and import language packs.

Dev and design coordination

Technical artists and designers should be already in contact at an early stage of the project to avoid designs which cannot be handled by a given platform. There should be regular meetings to talk about design and technical feasibility. Prototyping at an early stage of the project would be beneficial.

Product portfolio scaling with CGI Studio

If CGI Studio is used for several projects, there is the possibility to reuse some or maybe a lot of the previously developed objects depending on the project differences.
Reusable parts:

- Custom behaviors
  Customers create their own set of behaviors which cover functionality which is not available by default. This is common as customers bring in their own ideas and use cases. Along with delivered behaviors the custom behaviors can be reused for future projects.
- Custom behavior building block (BBB)
  Some behavior building blocks (combination of behaviors to get an extended functionality) are already part of CGI delivery, but customers can create their own BBBs to cover their use cases.
  BBBs can contain default behaviors but can also contain custom behaviors.
  Along with delivered BBBs the custom BBBs can be reused for future projects.
- Custom controls
  Beside the delivered set of controls, customers usually create their own set of Controls which cover their use cases. The delivered Controls are usually taken as samples and extended by additional functionality to cover customer use cases.
  Also completely new Controls are designed by customers which can also be reused in future projects along with the delivered controls.

Reusage of Controls and BBBs can be done either by creating a SCL (Sharable Component Library) which contains the elements needed and can be linked into every new project solution. In this case the elements cannot be altered and stay as intended in the previous project.
On the other hand, Controls and BBBs can be imported from a previous solution into the solution of a new project which provides the possibility to change the previous implementation.
Also, a combination of both ways is possible as the elements in the SCL can be cloned and the new, cloned element, can then be changed.

Revision #2
Created 2026-03-26 00:53:56 UTC by Kanai Tomoaki
Updated 2026-04-07 04:49:35 UTC by Kanai Tomoaki